

# The `phylo4` S4 classes and methods

Peter Cowan

December 21, 2007

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Definitions/slots</b>	<b>1</b>
2.1	<code>phylo4</code> . . . . .	1
2.2	<code>phylo4d</code> . . . . .	2
2.3	<code>multiphylo4</code> . . . . .	2
<b>3</b>	<b>Validity checking</b>	<b>2</b>
<b>4</b>	<b>Hacks/backward compatibility</b>	<b>2</b>
<b>5</b>	<b>To do/problems</b>	<b>3</b>

## 1 Introduction

This document describes the new `phylo4` S4 classes and methods, which are intended to provide a unifying standard for phylogenetic data in R. The base `phylo4` class is modeled on the the `phylo` class in `ape`. `phylo4d` and `multiphylo4` extend the `phylo4` class to include data or multiple tree respectively.

## 2 Definitions/slots

### 2.1 `phylo4`

Like `phylo`, the main components of the `phylo4` class are:

**edge** an  $N \times 2$  matrix of integers, where the first column ...

**edge.length** numeric list of edge lengths (length  $N$  or empty)

**Nnode** integer, number of nodes

**tip.label** character vector of tip labels (required)

**node.label** character vector of node labels (maybe empty)

**root.edge** integer defining root edge (maybe NA)

We have defined basic methods for **phylo4**: **show**, **print** (copied from **print.phylo** in **ape**), and a variety of accessor functions (see help files).

**summary** does not seem to be terribly useful in the context of a “raw” tree, because there is not much to compute: **end users?**

Print method: add information about (ultrametric, scaled, polytomies (zero-length or structural))?

## 2.2 phylo4d

The **phylo4d** class extends **phylo4** with data. Tip data, (internal) node data, and edge data are stored separately, but can be retrieved together or separately with **tdata(x, "tip")** or **tdata(x, "all")**.

edge data can also be included — is this useful/worth keeping?

## 2.3 multiphylo4

## 3 Validity checking

- number of rows of edge matrix ( $N$ ) == length of edge-length vector (if  $> 0$ )
- (number of tip labels)+(nNode)-1 ==  $N$
- data matrix must have row names
- row names must match tip labels (if not, spit out mismatches)
- 

Default node labels:

## 4 Hacks/backward compatibility

Hilmar Lapp very kindly showed a way to hack the **\$** operator so that it would provide backward compatibility with code that is extracting internal elements of a **phylo4**. The basic recipe is:

```
> setMethod("$", "phylo4", function(x, name) {  
+   attr(x, name)  
+ })
```

but this has to be hacked slightly to intercept calls to elements that might be missing. For example, **ape** detects whether log-likelihood, root edges, node labels, etc. are missing by testing whether they are **NULL**, whereas missing items are represented in **phylo4** by zero-length vectors in the slots (or **NA** for the root edge) — so we need code like

```
> if (!hasNodeLabels(x)) NULL else x@node.label
```

to handle these cases.

## 5 To do/problems

- Conflict with `nTips` if `ape` is loaded first: ask EP to get rid of this (obsolete?) function? (`Ntips` is the real `ape` function for getting the number of tips)
- basic tree manipulation: tip-dropping, `na.omit`, etc. — especially for multi-tree and tree-with-data cases
- tree-manipulation code: tree traversal (store current position as an attribute), pruning, etc.
- restrict/specify edges matrix to be integer?